



THIS PAGE IS
INTENTIONALLY
LEFT BLANK.

Kazimir Majorinc

Programski jezik LISP

Operacije i aplikacije

Povijest Lispa 24.



Razmjena vještina

Hacklab u mami

9. ožujka 2013.

I. dio - Elementarni

1. Edmund Berkeley, LISP - A simple introduction

Dobar uvod, kakvog do tada nije bilo, ali ništa za nas.

2. Robert Saunders, LISP - on the programming system

Funkcionalni argumenti, **FUNCTION** i **FUNARG**.

Pozivi funkcija poput

```
(G (QUOTE ((A) (B) (C))  
    (QUOTE (LAMBDA(X)X))))).
```

Izazivaju probleme. Drugi **QUOTE** nije dovoljno dobar.

Promotrimo funkciju **G** definiranu sa:

```
(LAMBDA (L FN)
  (COND ((NULL L) NIL)
        (T (G (CAR L)
                (QUOTE (LAMBDA ()
                        (G (CDR L) FN)))))))
```

Izračunajmo (G (QUOTE ((A) (B) (C))) (QUOTE (LAMBDA(X)X))).

```
L: ((A) (B) (C))
FN: (LAMBDA (X) X)
```

Poziv

```
(G (CAR L) (QUOTE (LAMBDA ()
                    (G (CDR L) FN))))
```

```
L: (A)
FN: (LAMBDA () (G (CDR L) FN))
```

„and now we are in trouble.“

Rješenje je „FUNARG device“ koji postoji u Lispu 1.5. ali nije uopće opisan u priručniku.

```
(LAMBDA (L FN)
  (COND ((NULL L) NIL)
        (T (G (CAR L)
                (FUNCTION (LAMBDA ()
                          (G (CDR L) FN)))))))
```

Izračunajmo (G (QUOTE ((A) (B) (C))) (QUOTE (LAMBDA(X)X))).

```
L: ((A) (B) (C))
FN: (LAMBDA (X) X)
```

Poziv

```
(G (CAR L) (QUOTE (LAMBDA ()
                  (G (CDR L) FN))))
```

```
L: (A)
FN: (FUNARG (LAMBDA () (G (CDR L) FN))
      ((L . ((A) (B) (C))) (FN . (LAMBDA (X) X))))
```

3. Hart & Levin, 240 exercises and solutions

Zgodno, elementarno.

Evaluate the following:

19. `car[cons[A;B]]`
20. `cons[car[(A . B)];cdr[(A . B)]]`
21. `car[cdr[(A . (B . C))]]`
22. `eq[car[(A . B)];cdr[(C . B)]]`
23. `eq[cdr[(A . B)];cdr[(C . B)]]`
24. `eq[A;car[cons[car[(A . B)];C]]]`
25. `atom[cons[A;B]]`
26. `atom[cdr[car[(A . C) . B]]]`
27. `car[cdr[car[(A . (B . C)) . D]]]`

D. Variables

4. Elaine Gord, Notes on debugging Lisp programs

Trivijalno (paziti na broj zagrada, dvije različite funkcije ne smiju imati isto ime i sl.)

5. Fischer Black, Styles of programming in Lisp

Za ili protiv „program feature.“

member[A; (C D A B)] = T

member[E; (C D A B)] = F

```
member[x;y]= prog [ [ ];  
A [ null[y]→ return[F];  
  equal[ x; car[y] ]→ return[T] ]  
  y := cdr[ y ]  
  go[ A ] ]
```

```
member[x;y]=[ null[ y ]→ F;  
  equal[ x; car[y] ]→ T  
  T → member[ x;cdr[y] ] ]
```

Za ili protiv funkcionala.

$$\text{ydot}[(A\ B\ C\ D); X] = ((A \cdot X)\ (B \cdot X)\ (C \cdot X)\ (D \cdot X))$$

$$\text{ydot}[x; y] = \text{maplist}[x; \lambda[j] \text{cons}[\text{car}[j]; y]]$$

$$\text{ydot}[x; y] = [\text{null}[x] \rightarrow \text{NIL} ; \\ \text{T} \rightarrow \text{cons}[\text{cons}[\text{car}[x]; y] \\ \text{ydot}[\text{cdr}[x]; y]]]$$

Primjedba (KM) Prednost funkcionala dolazi do izražaja u izrazima u kojima se upotrebljavaju kao varijable, ne kao konstante.

`maplist[x; f]`